REMARKS/ARGUMENTS

Claims 1-19 are pending in the application. Claims 1-5 are rejected under 35 U.S.C. § 101. Claims 6-7 are rejected under 35 U.S.C. § 112, second paragraph. Claims 1-5 and 8-14 are rejected under 35 U.S.C. §102(e) as being taught by Kyker et al., U.S. Patent No. 6,578,138 ("Kyker"). Claims 6-7 and 15-19 are rejected under 35 U.S.C. §103(a) as being unpatentable over Kyker et al., as applied to claims 5 and 14 above, and further in view of Rotenberg et al.'s "A Trace Cache Microarchitecture and Evaluation" IEEE ©1999 ("Rotenberg"). Claims 6-7 are amended to put the claims into better form; no new matter is added.

With regard to the §101 rejection, Applicants submit the claim 1 describes an apparatus; specifically, it describes a cache comprising a cache line to store an instruction segment further comprising a plurality of instructions stored in sequential positions of cache line in reverse program order. One of ordinary skill in the art will readily understand the useful and tangible results the embodiment described in claim 1 may be directed by, among other things, reading the specification. Therefore, Applicants submit claim 1 in its present form meets the requirements of 35 U.S.C. §101, and the current rejection of claims 1-5 should be withdrawn. *See* MPEP §706.03(a).

With regard to the §112 rejection of claims 6-7, Applicants submit claims 6-7 overcome the current rejections.

Applicants submit the cited references do not teach, suggest or disclose at least "[a] cache comprising: a cache line to store an instruction segment further comprising a plurality of instructions stored in sequential positions of the cache line in reverse program order" (*e.g.*, as described in claim 1).

The Office Action asserts Kyker teaches the relevant limitations at column 2, line 59 – column 3, line 33, including Figure 1 and Figure 2. *See* Office Action dated 4/18/2007, paragraph 11. Applicants disagree.

The first paragraph of the cited section, column 2, line 59 – column 3, line 33, states:

> FIGS. 1 and 2 illustrates an exemplary embodiment of a method of unrolling loops within a trace cache according to the present invention. In particular, this first exemplary method is applied to two exemplary traces T1 and T2 (illustrated in FIG. 2), which will be utilized throughout the description for convenience. Traces T1 and T2 represent traces built according to known methods described above, and will be contrasted with traces T1' and T2', which are built according to exemplary embodiments of methods according to the present invention. Exemplary trace T1 includes a total of four micro-ops, three of which form a loop. Trace T1 first includes a trace head T.sub.H, followed by the head of the loop L.sub.H. The third micro-op of trace T1 is the second micro-op of the loop, L.sub.2, and the fourth micro-op of the trace is the third micro-op of the loop, L.sub.3. The final micro-op L.sub.3 includes, for example, the backward taken branch whose target address is the head of the loop L.sub.H. The second exemplary trace T2 includes the same loop, L.sub.H, L.sub.1, L.sub.2, but does not include any micro-op preceding the loop itself. Accordingly, L.sub.H is also the trace head (and is therefore illustrated as T.sub.H /L.sub.H), followed by L.sub.2 and L.sub.3 (see FIG. 2).

The first paragraph of the cited section describes the "unrolling" of the illustrated traces T1 and T2 of Figures 1 and 2. Trace T1includes four elements – a trace head T.sub.H, and three elements that form the "loop": L.sub.H. L.sub.2, and L.sub.3. The second trace T2 includes the same loop, L.sub.H, L.sub.1, L.sub.2, but does not include any micro-op preceding the loop itself. In essence, the first paragraph describes the content and structure of the traces T1 and T2 to be "unrolled", but does not describe the unrolling process itself.

The second paragraph of the cited section states:

> In the first exemplary method, the trace cache does not end the current trace (i.e. the trace being built) at the occurrence of the backward taken branch, as may be done in a conventional trace cache. Rather, when the trace cache determines that a loop is present, the trace cache continues to build the trace by building additional

> iterations of the loop until the trace is a minimal length, for example until the trace is greater than twelve micro-ops long. In other words, the trace cache builds the loop repeatedly until the trace is, for example, over two trace lines long. In an exemplary embodiment, the existence of a loop is established by checking for the existence of a backward-taken branch. It should be understood, however, that the determination that a loop exists need not be an active determination. Rather, the phrasing is simply used to designate a general situation in which a loop exists. If no loop exists for a given micro-op, the trace cache may continue to build the trace as usual, i.e., according to given parameters and limitations (such as length, etc.) which may be incorporated in a trace cache which does not unroll loops according to the present invention.

The second paragraph of the cited section describes the "unrolling" process of Kyker in that when the trace cache determines that a loop is present, the trace cache continues to build the trace by building additional iterations of the loop until the trace is a minimal length. In short, the trace cache *builds a loop.* The cited section further describes that the trace cache *builds a loop* by checking for the existence of a backward taken branch. Kyker defines a backward taken branch as follows: "[a] backward taken branch generally occurs when the target address of a branch is a prior micro-op, and in particular, for purposes of this description, a prior micro-op of the trace." *See* column 1, lines 30-33. Therefore, while building a loop, a backward taken branch may be utilized by determining that the target address of the following branch is a prior micro-op, *thereby enabling the building of the loop.*

Applicants submit the cited section does not teach or suggest a cache comprising a cache line to store an instruction segment further comprising a plurality of instructions stored in sequential positions of the cache line in reverse program order (*e.g.*, as described in claim 1). Indeed, reverse program order is not discussed at all, even generally. One of ordinary skill in the art will readily understand that a trace cache that builds a loop is not the same as a cache that stores a plurality of instructions stored in sequential positions of the cache line in reverse

program order. Applicants submit the Kyker reference fails to teach or suggest at least the above-discussed limitations of claim 1, and therefore the current §102 rejection of claim 1 is lacking and should be withdrawn.

Rotenberg fails to make up for the deficiencies of Kyker. Rotenberg is directed to trace cache sequencing, selection, and prediction. It does not teach at least "[a] cache comprising: a cache line to store an instruction segment further comprising a plurality of instructions stored in sequential positions of the cache line in reverse program order" (*e.g.*, as described in claim 1).

As the cited references fail to teach or suggest each and every limitation of claim 1, they fail to support proper §102 or §103 rejections. Applicants submit claim 1 is allowable. Independent claims 5, 6, 8 and 14 contain similar allowable limitations, and are therefore allowable for similar reasons. Claims 2-4, 7, 9-13 and 15-19 are allowable for depending from allowable base claims.

For at least all the above reasons, the Applicant respectfully submits that this application is in condition for allowance. A Notice of Allowance is earnestly solicited.

The Examiner is invited to contact the undersigned at (408) 975-7500 to discuss any matter concerning this application. The Office is hereby authorized to charge any additional fees or credit any overpayments under 37 C.F.R. §1.16 or §1.17 to Deposit Account No. **11-0600**.

Respectfully submitted,

KENYON & KENYON LLP

Date: July 18, 2007 By: /Sumit Bhattacharya/
Sumit Bhattacharya
(Reg. No. 51,469)

KENYON & KENYON LLP
333 West San Carlos St., Suite 600
San Jose, CA 95110
Telephone: (408) 975-7500
Facsimile: (408) 975-7501

Customer No. 25693